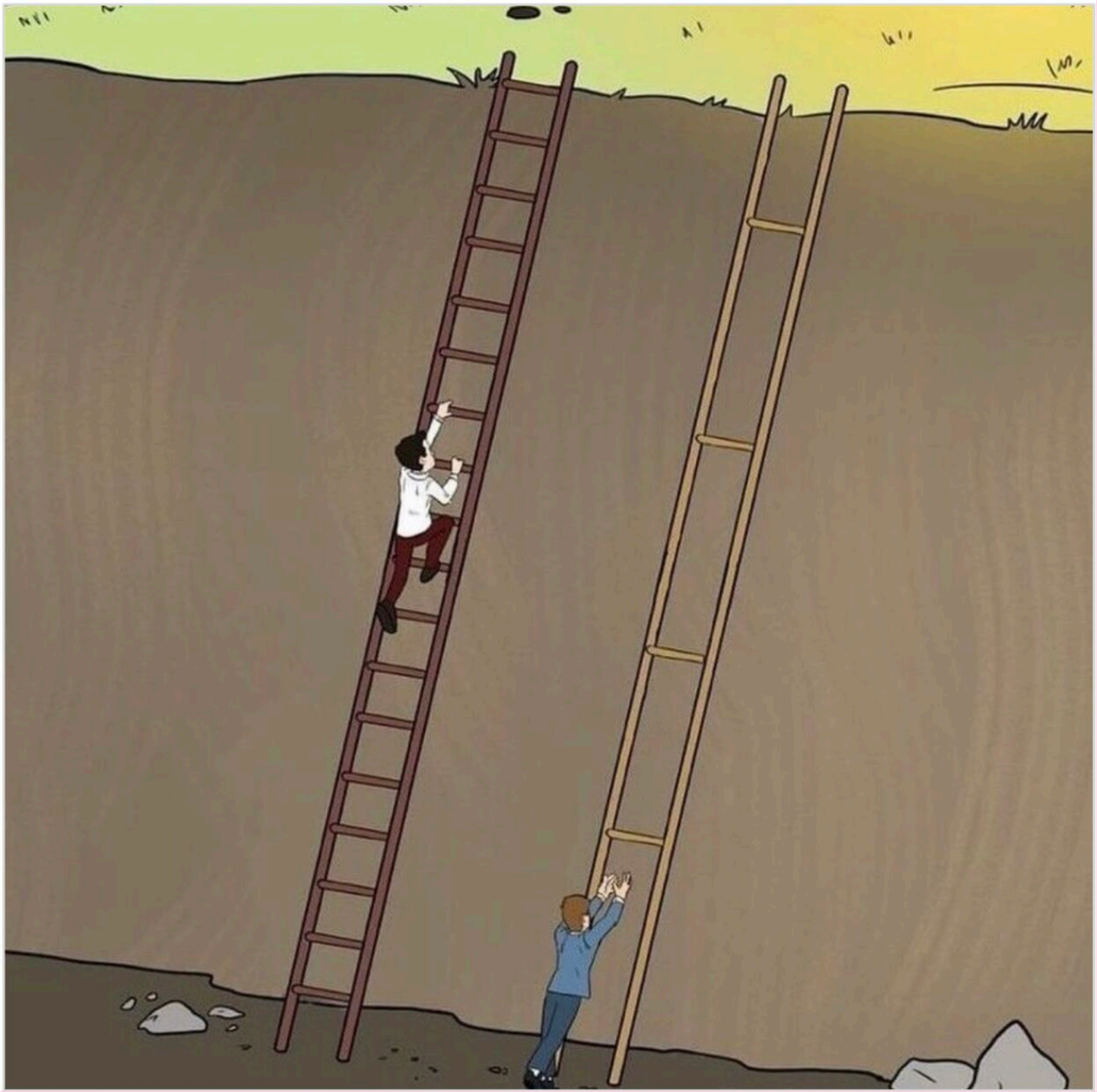
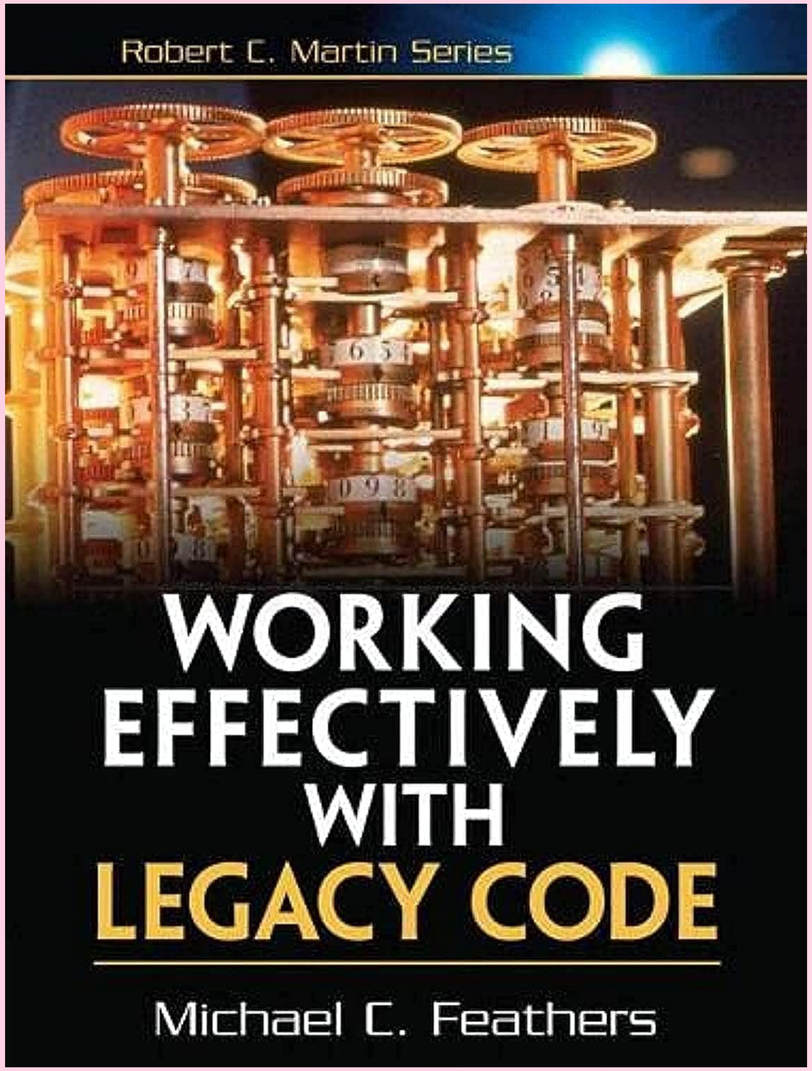
KODY FINTAK

Resources



Biting more than you can chew

“Programming is the art of doing one thing at a time”
- **Michael Feathers**



Make it smaller

Release

Story

Deployment

Integration

Commit

Code

Make it smaller

Release

Story

Deployment

Integration

Commit

Code

micro-commits

“one small step”

"a tiny commit" that does "one tightly-scoped change".

micro-commits

“one small step”

"a tiny commit" that does "one tightly-scoped change".

“extract method totalTax on Cart”

micro-commits

“one small step”

"a tiny commit" that does "one tightly-scoped change".

“extract method totalTax on Cart”

“add gap spacing between items on cart”

micro-commits

“one small step”

"a tiny commit" that does "one tightly-scoped change".

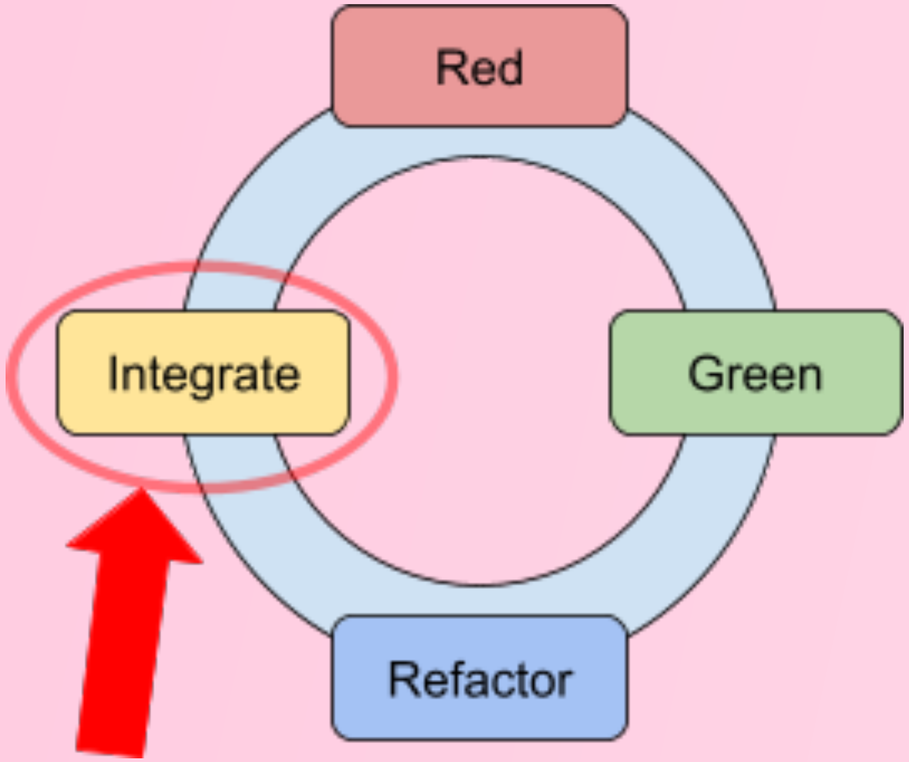
“extract method totalTax on Cart”

“add gap spacing between items on cart”

“-r rename variable subtotal”

micro-commits

“one small step”




TURN-BASED
Astarion
10/10
N
X:-298 Y:-338
AP
EXIT
BA
Shadowheart receives status Difficult Terrain
Gale receives status Difficult Terrain
Lae'zel receives status Difficult Terrain


TURN-BASED
Astarion
10/10
SAVE YOUR GAME
X:-298 Y:-338
Shadowheart receives status Difficult Terrain
Gale receives status Difficult Terrain
Lae'zel receives status Difficult Terrain
AP
EXIT
BA


git reset —hard

micro-commits

“one small step”

Every commit is **deployable**

micro-commits

“one small step”

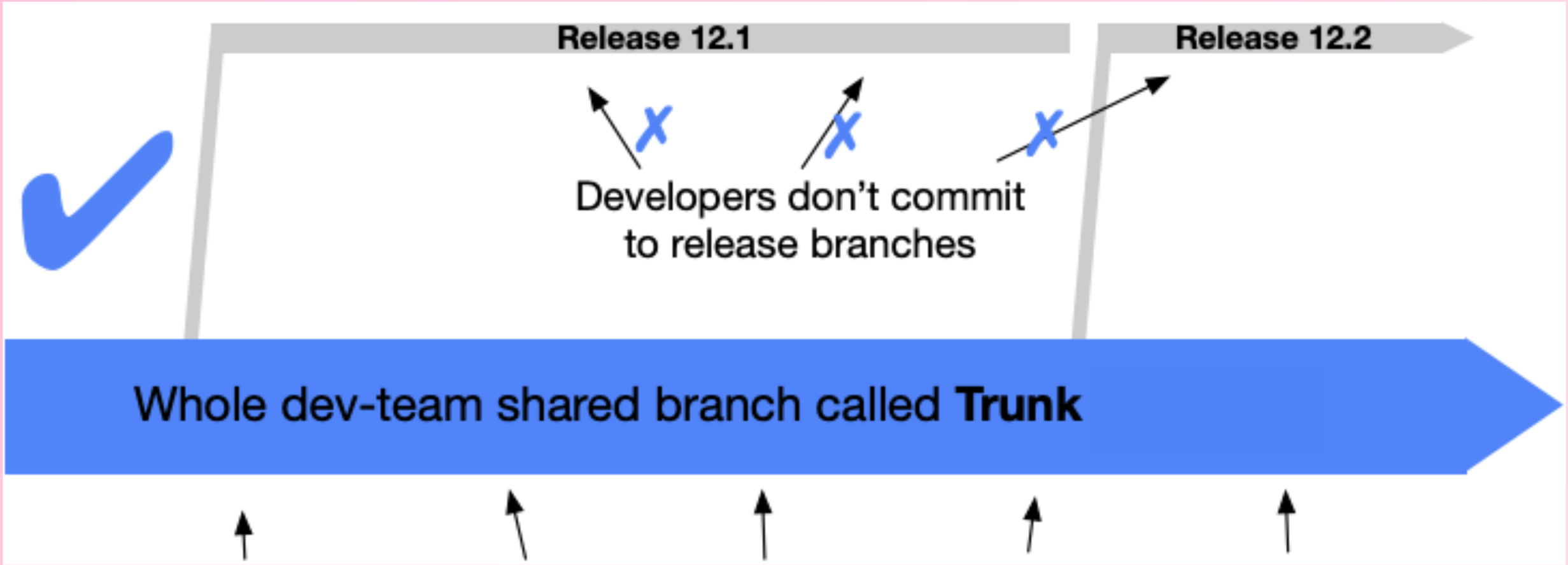
Every commit is **deployable**

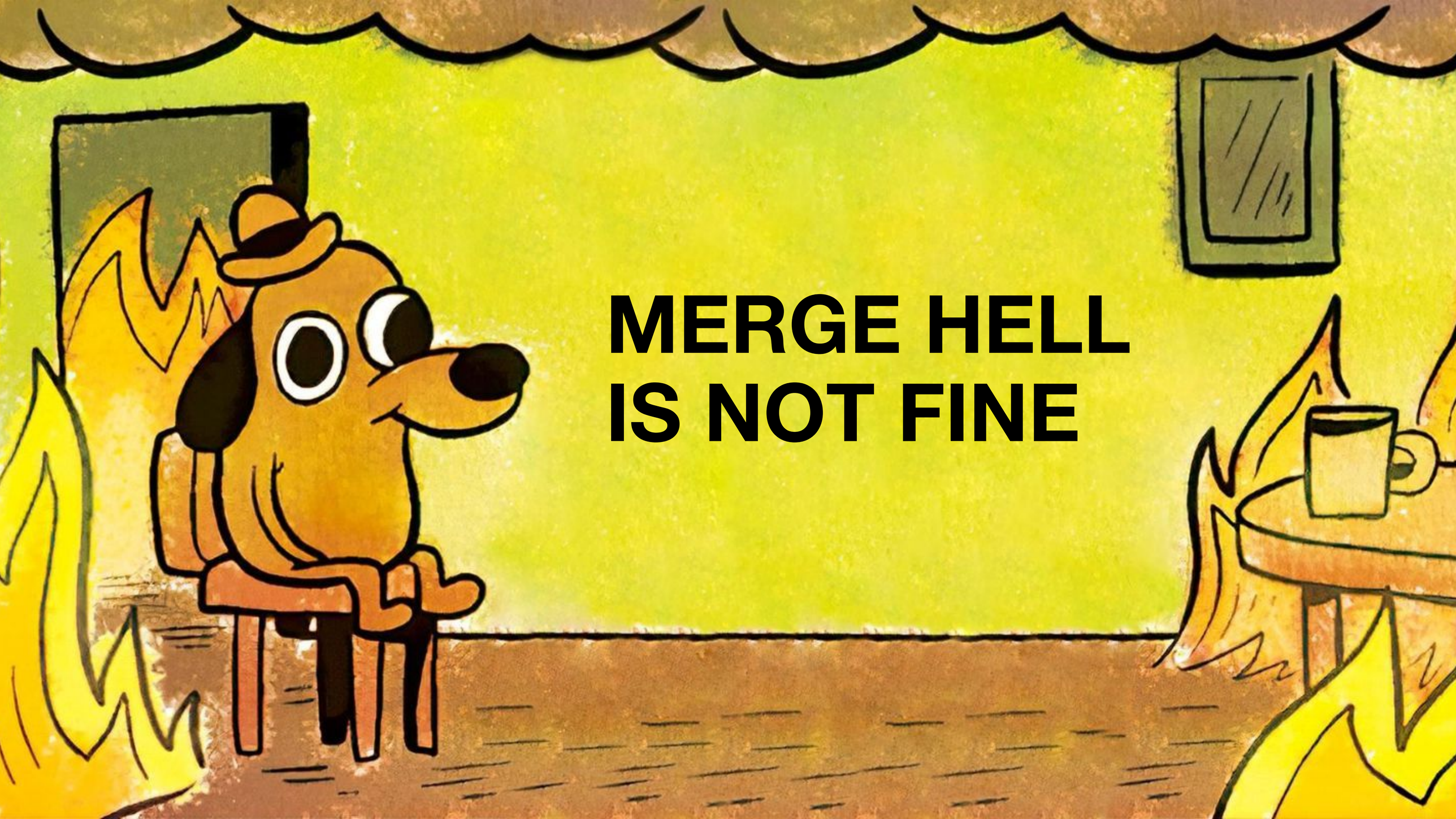
Separate deployability with releasability

Only make changes visible to users when you are ready to release

Feature Flags

Trunk-Based Development




MERGE HELL
IS NOT FINE

Continuous Integration

“Development teams use CodeOwnership to minimize conflicts among people editing. The longer engineers hold on to modules, the more important it is to minimize conflicts.

What if engineers didn't hold on to modules for more than a moment? What if they made their (correct) change, and *presto!* everyone's computer instantly had that version of the module?

You wouldn't ever have IntegrationHell, because the system would always be integrated. You wouldn't need CodeOwnership, because there wouldn't be any conflicts to worry about.”

- **c2 wiki**

What is the hardest part of TDD?

What is the hardest part of TDD?

Figuring out what the first failing test

What is the hardest part of working in small steps?

What is the hardest part of working in small steps?

Figuring out the next small step to take

- Are we only working on one thing?
- Is there a smallest step we can take?
- Is there a way we can make this change deployable?


mob

Carl

Maria

Carl

Maria

```
# create mob session
mob start
mob start -b cart-total
```

Carl

```
# create mob session
mob start
mob start -b cart-total
```

```
# rotate
mob next
```

Maria

Carl

```
# create mob session
mob start
mob start -b cart-total
```

```
# rotate
mob next
```

Maria

```
# next typist
mob start
mob start -b cart-total
```

Carl

```
# create mob session
mob start
mob start -b cart-total
```

```
# rotate
mob next
```

Maria

```
# next typist
mob start
mob start -b cart-total
```

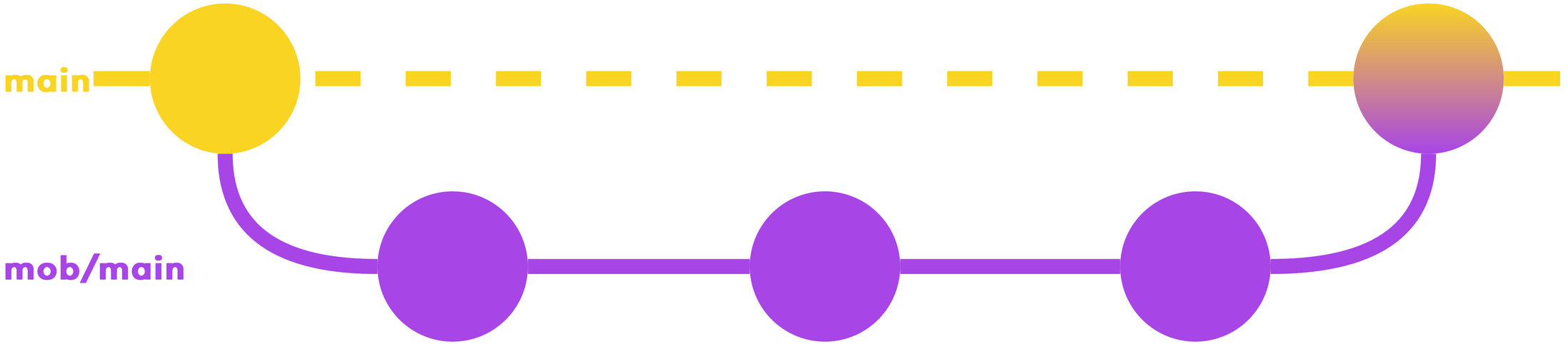
```
# done - ready to commit
mob done
git commit -m "extract method totalTax on Cart"
```

what if we want to commit a change before the 1st rotation timer is up?

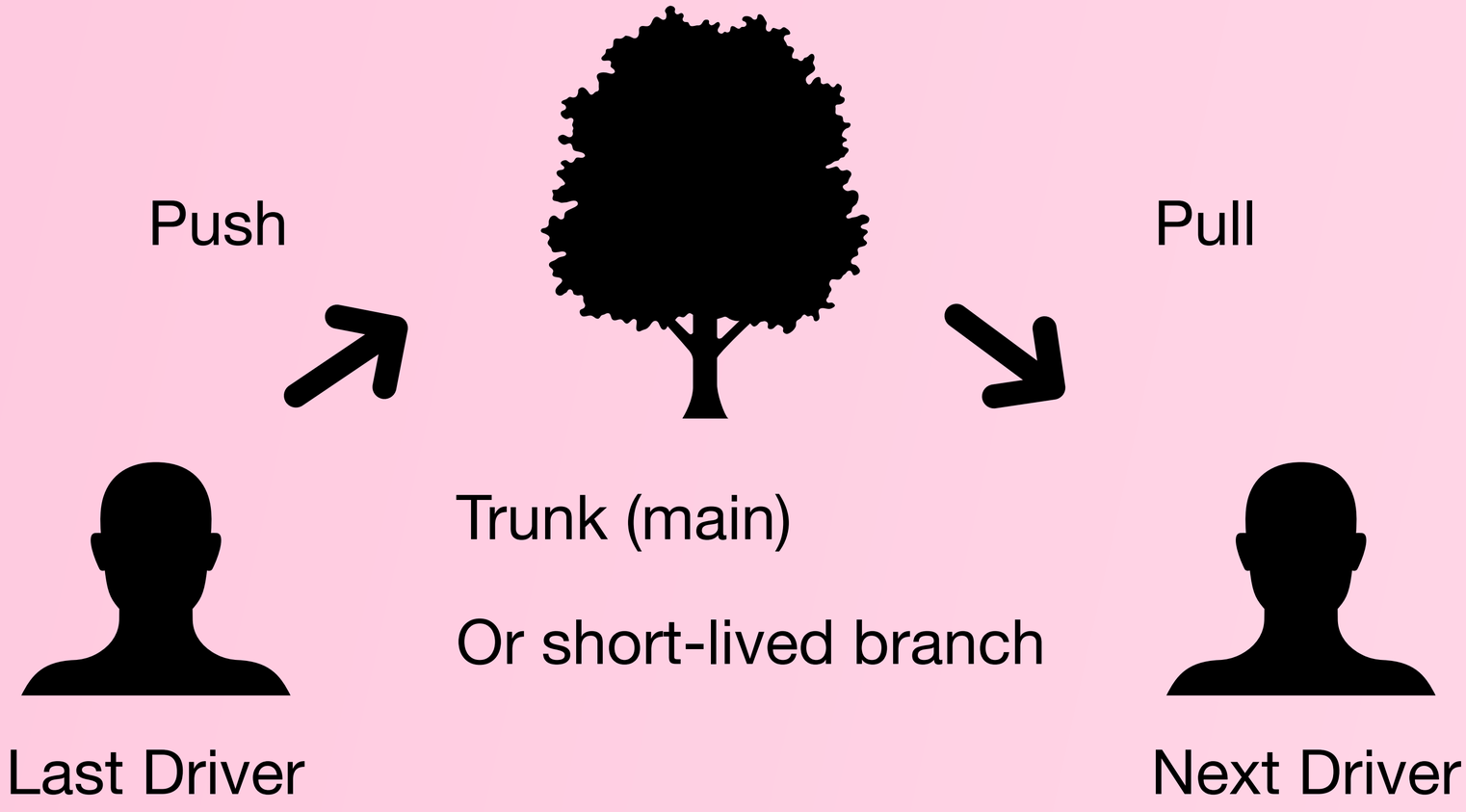

```
# create mob session
mob start
mob start -b cart-total
```


```
# create mob session
mob start
mob start -b cart-total
```

```
# done - ready to commit
mob done
git commit -m "-r rename variable subtotal"
```


main
mob/main

Git Handoff


Push
Pull
Trunk (main)
Or short-lived branch
Last Driver
Next Driver

Git Handoff

- Driver types some code, commits and pushes to trunk.
- Then decide should we rotate or keep going?

- Hmmmmmmmmmmmm?

Git Handoff

- Driver types some code, commits and pushes to trunk.
- Then decide should we rotate or keep going?
- Hmmmmmmmmmmmmm?

Hmmmmmmmmmmmmmm?

Write a Test.

Make it Pass.

Integrate.

Rotate.

Write a Test.

Make it Pass.

Integrate.

Rotate.

Refactor.

Integrate.

Rotate.

Write a Test.

Make it Pass.

Integrate.

Rotate.

Refactor.

Integrate.

Rotate.

Refactor.

Integrate.

Rotate.

Trunk-Based Ping-Pong

we (***can)*** rotate anytime we can **commit and push** our changes to Trunk

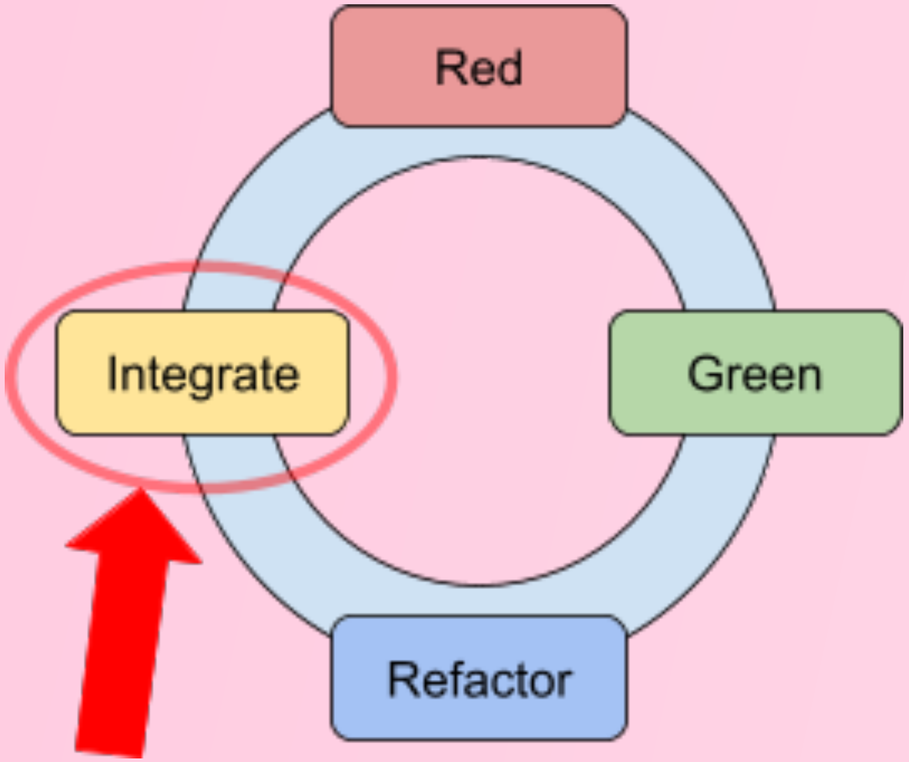
Trunk-Based Ping-Pong

we (***can)*** rotate anytime we can **commit and push** our changes to Trunk

This assumes we are working in small-steps

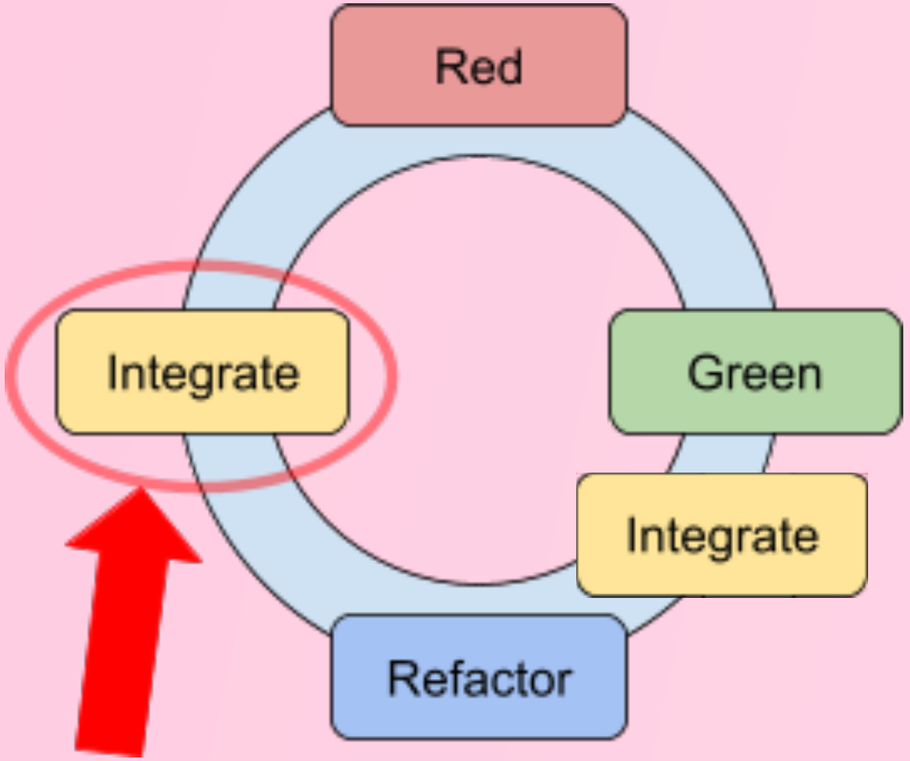
Trunk-Based Ping-Pong

In a TDD Workflow



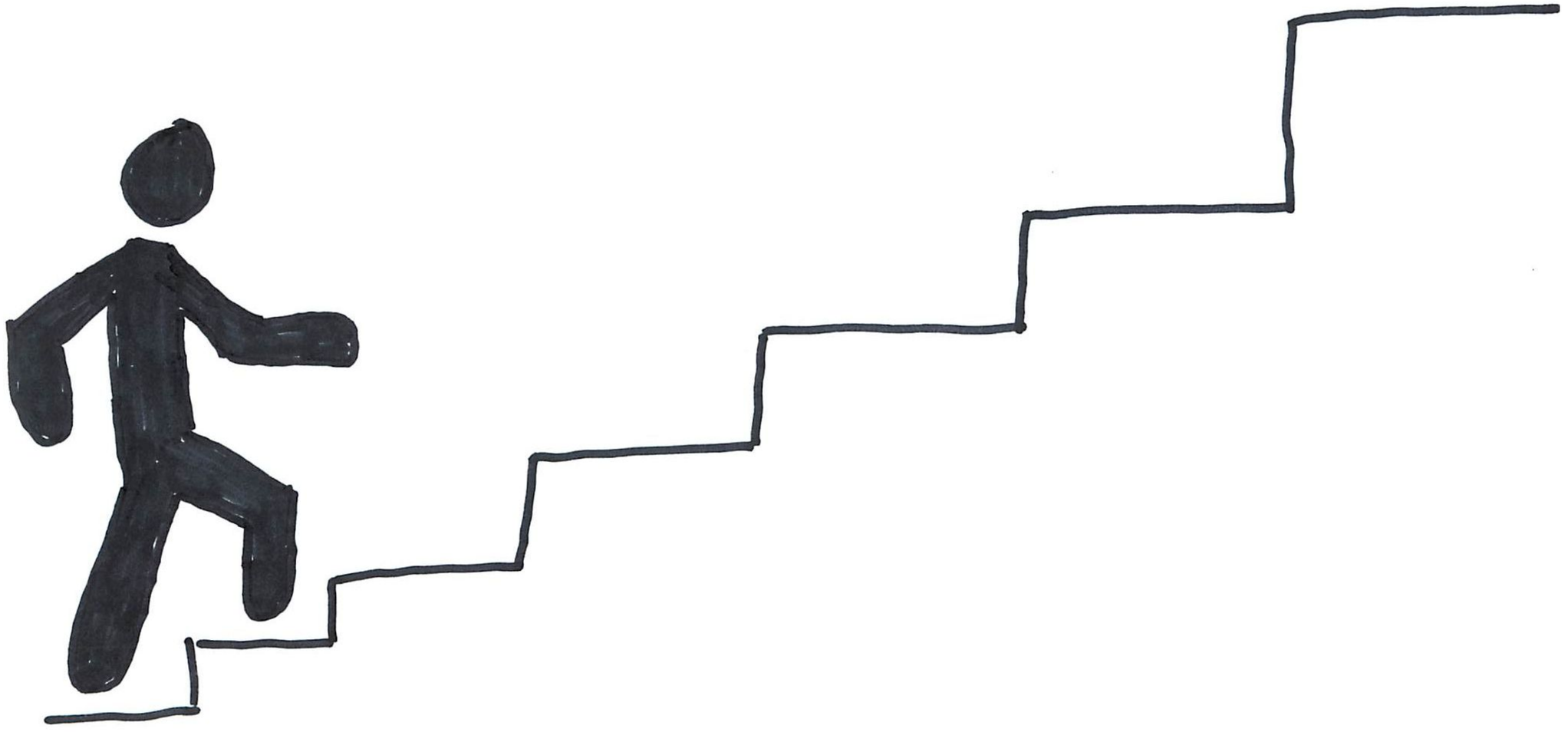
Trunk-Based Ping-Pong

In a TDD Workflow



How many step(s) before rotating?

"ONE SMALL STEP"



Exercise: FizzBuzz


1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
Fizz Buzz
16

What did we think?

Rotation Style 1

mob.sh

Pros:

- Easy to rotate
- Consistent rotation
- Easier to take consistent breaks

Cons:

- Extra work to push to trunk
- Because it’s easy to rotate, easy to increase WIP
- Longer the WIP branch, the longer we haven’t pulled from trunk

Rotation Style 2

Git Handoff

Pros:

- Easy to push to trunk
- Incentivized to lower WIP to keep rotation easy
- Easier to keep up to date with trunk

Cons:

- Harder to rotate
- No consistency for rotation
- Remembering to take breaks is hard

Rotation Style 3

Trunk Based Ping Pong

Pros:

- Easy to push to trunk
- Constantly rotating
- Easy to keep up to date with trunk
- Fun
- Easy to rotate (If large WIP…. git reset —hard)

Cons:

- Remembering to take breaks is still hard
- Can be inefficient if the cost of rotating is expensive
- Need to be able to take small steps

TDD Ping-Pong

TDD Ping-Pong

- Person A writes a Failing unit test (red)

TDD Ping-Pong

- Person A writes a Failing unit test (red)
- Person B makes it pass

TDD Ping-Pong

- Person A writes a Failing unit test (red)
- Person B makes it pass
- Person B writes the next failing test

TDD Ping-Pong

- Person A writes a Failing unit test (red)
- Person B makes it pass
- Person B writes the next failing test
- Person A makes it pass

TDD Ping-Pong

- Person A writes a Failing unit test (red)
- Person B makes it pass
- Person B writes the next failing test
- Person A makes it pass
- **Either developer can refactor as much as they want when the tests are passing**

TDD Ping-Pong

We rotate on ***Red***

Trunk-Based Ping-Pong

We rotate on ***Green***

working collaboratively <——> working in small-steps

Thank You!

LinkedIn: @Kody Fintak

Link-to-the-resource

end